

**Fondamenti di Informatica L-A (A.A. 2005/2006) - CdS Ingegneria Informatica**  
**Prof.ssa Mello & Prof. Bellavista – I Prova Intermedia del 02/11/2005 - durata 2h30m**  
**COMPITO F**

**ESERCIZIO 1 (12 punti)**

Un centro di meteorologia vuole effettuare in modo automatizzato statistiche sull'umidità registrata quotidianamente nell'arco di un mese. In particolare, vuole ottenere il tasso di umidità media (non considerando il giorno più umido del mese in esame) e la data del giorno più umido col relativo tasso di umidità. A tale scopo si realizzi:

1) una funzione

```
int mediaUmidita(float umidita[], int length, float* maxUmiditaTasso, int* maxUmiditaGiorno)
```

che noto il numero di giorni presi in esame **length** e i tassi di umidità **umidita**, restituisca l'umidità media (escludendo dal calcolo il giorno più umido) come **int**. Inoltre, tramite i parametri **maxUmiditaTasso** e **maxUmiditaGiorno** la funzione deve restituire rispettivamente il tasso di umidità e la data del giorno più umido. Si assuma che il giorno 0 corrisponda al primo giorno del mese, il giorno 1 al secondo e così via; in relazione al parametro **umidita** si noti che l'umidità presente in **umidita[0]** corrisponde al primo giorno del mese, l'umidità in **umidita[1]** al secondo e così via; **(7 punti)**

2) un programma **main()** che

a) chieda all'utente il numero di giorni **G** del mese preso in esame e controlli che **G** abbia valore tra 28 e 31, estremi inclusi. In caso contrario, si richieda nuovamente il numero di giorni;

b) chieda all'utente di inserire **G** tassi di umidità in ordine dal primo all'ultimo giorno del mese, controllando che i valori inseriti siano tra 0 e 100 compresi; altrimenti si richieda nuovamente l'inserimento del valore di umidità;

c) richiami opportunamente la funzione **mediaUmidita(...)**;

d) stampi l'umidità media restituita da **mediaUmidita (...)**, il tasso di umidità e la data del giorno più umido. **(5 punti)**

**ESERCIZIO 2 (6 punti)**

Il seguente programma C compila correttamente? In caso affermativo, quali sono i valori stampati a tempo di esecuzione? (Si motivi opportunamente la risposta data)

```
#include <stdio.h>

#define DIM 4

void fun(char *x1, char* x2, int dim){
    int i;
    char temp;
    for(i=dim-1;i>=0;i--){
        if(*(x1+i)<=*(x2+dim-1-i)){
            temp=x1[i];
            x1[i]=x2[dim-1-i];
            x2[dim-1-i]=temp;
        }
    } return;
}

int main(){
    int i=0;
    char c1[DIM]={'a','b','c','d'};
    char c2[DIM]={'b','d','a','c'};

    fun(c1,c2,DIM);

    for(;i<DIM;i++) printf("%c ",c1[i]);
    printf("\n");
    for(;i>0;i--) printf("%c ",c2[DIM-i]);
    printf("\n");
    return 0;
}
```

### ESERCIZIO 3 (6 punti)

Si scriva una funzione iterativa `char fun(char *a, char *b)` che, ricevuti come parametri in ingresso due stringhe ben formate `a` e `b`, restituisca come valore il carattere di `a` che compare più volte in `b` (si assuma che in `a` un carattere possa comparire al più una volta). Ad esempio, la chiamata `fun("zona", "pazzesco")` deve restituire 'z'.

Si proponga una possibile funzione chiamante.

### ESERCIZIO 4 (4 punti)

Data la funzione:

```
int fun(int a, int b, int* c){
    *c=*c+b%a;
    if(a>b) return 1;
    else {
        b=b/a;
        return a*fun(a,b,c);
    }
}
```

e la funzione chiamante:

```
int main(){
    int r=0;
    fun(4,30,&r);
    return 0;
}
```

mostrare la sequenza dei record di attivazione.

### ESERCIZIO 5 (2 punti)

Un elaboratore rappresenta i numeri interi su 8 bit dei quali 7 sono dedicati alla rappresentazione del modulo del numero e uno al suo segno. Indicare come viene svolta la seguente operazione aritmetica e determinarne il risultato traslandolo poi in decimale per la verifica:

**28-74**

### ESERCIZIO 6 (2 punti)

Date le definizioni: `char s1[]="pippo", *s2;`

le seguenti istruzioni sono corrette? In caso affermativo, si dica che cosa viene stampato a video, motivando opportunamente la risposta.

```
s2 = s1; *(s2+1) = 'z'; printf("%s\n", s1);
```

# Soluzioni

## Esercizio 1

```
#include <stdio.h>
#define DIM 31

int mediaUmidita(float umidita[], int length, float* maxUmiditaTasso, int*
maxUmiditaGiorno){
    int i; float somma=0;
    *maxUmiditaGiorno=0;
    *maxUmiditaTasso=umidita[0];

    for(i=1;i<length;i++){
        if(umidita[i]>*maxUmiditaTasso){
            somma=somma+*maxUmiditaTasso;
            *maxUmiditaTasso=umidita[i];
            *maxUmiditaGiorno=i;
        }
        else somma=somma+umidita[i];
    }
    return (int)(somma/((float)(length-1)));
}

int main(){
    float umidita[DIM];
    float maxUmiditaTasso;
    int maxUmiditaGiorno, media, giorni, i;

    do { printf("Numeri giorni?\n");
        scanf("%d",&giorni);
    } while((giorni<28)|| (giorni>31));

    for(i=0;i<giorni;i++){
        do { printf("Umidita' giorno %d?\n",i);
            scanf("%f",&umidita[i]);
        } while((umidita[i]<0)|| (umidita[i]>100));
    }
    media=mediaUmidita(umidita, giorni, &maxUmiditaTasso, &maxUmiditaGiorno);
    printf("media %d; più umido %f il %d\n",media,maxUmiditaTasso,maxUmiditaGiorno);
    return 0;
}
```

## Esercizio 2

c b d d

b c a a

La funzione fun() scorre gli array x1 e x2; il primo dalla fine all'inizio, il secondo dall'inizio alla fine. Mentre scorre i due array, fun() confronta i valori di x1 ed x2; se il valore di x1 è minore o uguale a quello di x2, inverte i valori di x1 ed x2.

La funzione main() stampa gli array c1 e c2 modificati dall'inizio alla fine.

## Esercizio 3

```
char fun(char* a, char* b){
    int occCharCorrente, occCharC=0, i=0, j;
    char charC='';

    while(a[i]!='\0'){
        j=0;
        occCharCorrente=0;
        while(b[j]!='\0'){
            if(a[i]==b[j]) occCharCorrente++;
            j++;
        }
    }
}
```

```

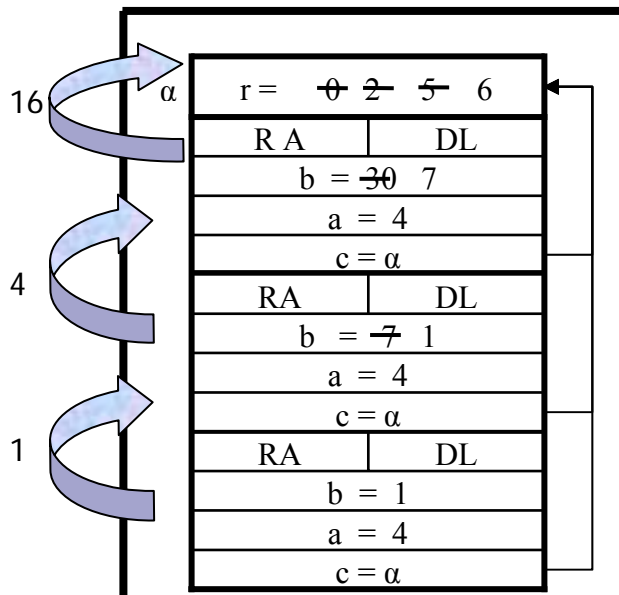
        if (occCharCorrente > occCharC) {
            occCharC = occCharCorrente;
            charC = a[i];
        }
        i++;
    }
    return charC;
}

int main() {
    char piuFrequente;
    char a[] = "abcdefghilmno";
    char b[] = "aabbdbdeh";
    piuFrequente = fun(a, b);
    printf("piuFrequente %c\n", piuFrequente);
    return 0;
}

```

**Esercizio 4**

La funzione effettua una divisione; come risultato restituisce 16, **r** assume il valore 6



**Esercizio 5**

Sottrazione dei moduli (numero maggiore meno numero minore)

```

1001010-   74-
0011100=   28=
-----
0101110    46

```

Aggiunta del segno

10101110

**Esercizio 6**

Le istruzioni sono corrette.  $s2 = s1$  assegna a  $s2$  il valore dell'indirizzo del primo elemento di  $s1$ .  $*(s2+1) = 'z'$  assegna al secondo elemento di  $s2$  il valore 'z'. Infine la `printf()` stampa su `stdout` `pzppp`, visto che dopo il primo assegnamento, le due variabili puntano allo stesso indirizzo di memoria.